EECS 583 – Class 15
Exam Review

University of Michigan

November 1, 2021

Announcements

- Research Paper Presentations
  - Each group sign up for 15 min slot on the EECS 583 calendar
  - Nov 8 – Dec 6: presentations during class
- Midterm Exam
  - Wednesday, Nov 3, Hybrid format
  - In person
    - 10:30am-12:15pm + 15 mins extra time for logistics
    - Questions answered in the hallway
  - Virtual
    - 10:30am-12:15pm + 30 mins extra time for logistics
    - Questions about exam can be posted on piazza and will be answered ASAP
  - Covers through modulo scheduling (last lecture), no register allocation

Research Paper Presentation Logistics

- Monday Nov 8 – Monday Dec 6
  - Signup for slot on Google calendar (just like project proposals)
  - Sign up for earliest slot available on the day you want to present → no gaps
  - Not all days will be full
- Each group: 15 min slot– You will be cut off if you go long!
  - Allow 2 mins for Q&A, so 13 min talk
  - Tag-team presentation – Divide up as you like but everyone must talk
  - Max of 15 slides (for the group)
  - Submit paper pdf 1 week ahead (sooner if possible so it can be posted)
  - Slides (pptx or pdf) night before

Research Paper Presentation Format

- Make your own slides!!!
  - Don’t just lift figures from the pdf (graphs/tables ok to lift)
  - Don’t have too many all text slides
  - No long sentences on slides, don’t just read the slides
- Points to discuss
  - Intro/Motivation – area + problem + why it being solved
  - How the technique works, examples are helpful
  - Some results, but don’t show 10 graphs
  - Commentary
    - What is best about the paper? Why is the idea so awesome? Don’t focus on results
    - What are limitations/weaknesses of the approach (be critical!)

Research Paper Presentation – Audience Members

- Research presentations != skip class
  - You should attend or watch the Zoom video
- Grading + give comments to your peers
  - Class + Yunjie/Ze & I will evaluate each group's presentation and provide feedback
  - Each person will submit evaluation sheet for the day's presentations
    - Online Google form
    - 24 hrs to submit
  - Yunjie & Ze will anonymize comments and email to each group
  - Be critical, but constructive with your criticisms
    - What was good about the talk, what could be improved.
    - Don't try to give separate comments for each group member, just evaluate the entire team

From Last Time: Homework Problem

latencies: add=1, mpy=3, ld = 2, st = 1, br = 1

```
for (j=0; j<100; j++)
  b[j] = a[j] * 26
```

```
      LC = 99
Loop: 1: r3 = load(r1)
      2: r4 = r3 * 26
      3: store (r2, r4)
      4: r1 = r1 + 4
      5: r2 = r2 + 4
      7: brlc Loop
```

How many resources of each type are required to achieve an II=1 schedule?

If the resources are non-pipelined, how many resources of each type are required to achieve II=1

Assuming pipelined resources, generate the II=1 modulo schedule.

Homework Problem – Answers in Red

latencies: add=1, mpy=3, ld = 2, st = 1, br = 1

```
for (j=0; j<100; j++)
  b[j] = a[j] * 26
```

```
      LC = 99
Loop: 1: r3 = load(r1)
      2: r4 = r3 * 26
      3: store (r2, r4)
      4: r1 = r1 + 4
      5: r2 = r2 + 4
      7: brlc Loop
```

How many resources of each type are required to achieve an II=1 schedule?

For II=1, each operation needs a dedicated resource, so: 3 ALU, 2 MEM, 1 BR

If the resources are non-pipelined, how many resources of each type are required to achieve II=1

Instead of 1 ALU to do the multiplies, 3 are needed, and instead of 1 MEM to do the loads, 2 are needed. Hence: 5 ALU, 3 MEM, 1 BR

Assuming pipelined resources, generate the II=1 modulo schedule.

See next few slides

Problem continued

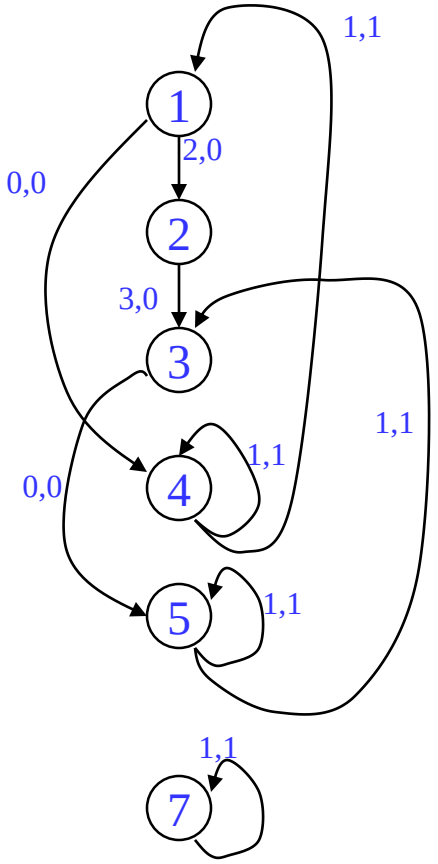
Assume II=1 so resources are: 3 ALU, 2 MEM, 1 BR

DSA converted code below (same as example in class)

LC = 99

```
Loop:  1: r3[-1] = load(r1[0])
       2: r4[-1] = r3[-1] * 26
       3: store (r2[0], r4[-1])
       4: r1[-1] = r1[0] + 4
       5: r2[-1] = r2[0] + 4
       remap r1, r2, r3, r4
       7: brlc Loop
```

Dependence graph (same as example in class)



RecMII = 1
RESMII = 1
MII = MAX(1,1) = 1

Priorities
1: H = 5
2: H = 3
3: H = 0
4: H = 4
5: H = 0
7: H = 0

Problem continued

resources: 3 alu, 2 mem, 1 br
latencies: add=1, mpy=3, ld = 2, st = 1, br = 1

LC = 99

Loop:

```
1: r3[-1] = load(r1[0])
2: r4[-1] = r3[-1] * 26
3: store (r2[0], r4[-1])
4: r1[-1] = r1[0] + 4
5: r2[-1] = r2[0] + 4
remap r1, r2, r3, r4
7: brlc Loop
```

Rolled Schedule

0	7 1 4 2 3 5

Unrolled Schedule

0	1 4	stage 1
1		stage 2
2	2	stage 3
3		stage 4
4		stage 5
5	3 5 7	stage 6
6		

Scheduling steps:

Schedule brlc at time II-1
Schedule op1 at time 0
Schedule op4 at time 0
Schedule op2 at time 2
Schedule op3 at time 5
Schedule op5 at time 5
Schedule op7 at time 5

MRT

	alu0	alu1	alu2	m1	m2	br
0	X	X	X	X	X	X

Problem continued

The final loop consists of a single MultiOp containing 6 operations, each predicated on the appropriate staging predicate. Note register allocation still needs to be performed.

```
    LC = 99
Loop:
    r3[-1] = load(r1[0]) if p1[0]; r4[-1] = r3[-1] * 26 if p1[2]; store (r2[0], r4[-1]) if p1[5]; r1[-1] = r1[0] + 4 if p1[0]; r2[-1] = r2[0] + 4 if p1[5]; brlc Loop
```

Exam Review

Virtual Exam Logistics (see piazza for more details)

- Wednesday Nov 3
- 10:30-12:15 + 30 mins for logistics
- Gradescope to distribute/collect exams, accessible via canvas
- Steps
  - Download, take exam, scan & submit
    - Print out and write on exam sheets
    - Just write answers on paper
    - Use electronic method (ie tablet) to create electronic answers
  - Exam itself should take < 2 hrs (1.5 hrs most likely)
  - So lots of slack time to deal with difficulties, but email course staff if you run into problems
- Use piazza to ask questions during the exam
  - We will answer ASAP. Be sure to read others questions before posting your own.

In-person Exam Logistics

- Wednesday Nov 3 – 1500 EECS
- 10:30-12:15 + 15 mins for logistics
- Printed exams available in classroom
- Steps – Normal pre-COVID exam
  - Exam itself should take < 2 hrs (1.5 hrs most likely)
  - Course staff will be outside 1500 EECS to answer questions

What to Expect

- Exam format
  - Open notes, open internet
  - Apply techniques we discussed in class
  - Reason about solving compiler problems – how/why things are done
  - A couple of thinking problems
  - No LLVM code
- Honor code and cheating
  - Must sign honor code acknowledging that you have neither given no received aid on the exam
  - Please do not share answers or talk to other students during the exam
  - Graduate class, so we don't expect cheating to be an issue
    - But we will investigate any anomalies that arise

Studying

- 5 exams (F12-F13, F18, F19, F20) are posted on the course website
  - Note – Past exams may not accurately predict future exams!!
  - Fomat and length will be similar
- Preparing yourself
  - Yes, you should study even though its open notes
    - Lots of material that you have likely forgotten from early this semester
    - Refresh your memories, especially the old topics
    - No memorization required, but you need to be familiar with the material to finish the exam
  - Go through lecture notes, especially the examples!
  - If you are confused on a topic, go through the reading
  - Go through the practice exams (Don't look at the answer) as the final step

Exam Topics

- Control flow analysis
  - Control flow graphs, Dom/pdom, Loop detection
  - Trace selection, superblocks
- Predicated execution
  - Control dependence analysis, if-conversion
- Dataflow analysis
  - Liveness, reaching defs, DU/UD chains, available defs/exprs
  - Static single assignment – Make sure you understand SSA!
- Optimizations
  - Classical: Dead code elim, constant/copy prop, CSE, LICM, induction variable strength reduction
  - ILP optimizations - unrolling, tree height reduction, induction/accumulator expansion – Just understand the concepts
  - Speculative optimization – like HW2

Exam Topics - Continued

- Acyclic scheduling
  - Dependence graphs, Estart/Lstart/Slack, list scheduling
  - Code motion across branches, speculation, exceptions
- Software pipelining
  - DSA form, ResMII, RecMII, modulo scheduling
  - Make sure you can modulo schedule a loop!
  - Execution control with LC, ESC
- Ignore topics we haven't covered (that may be on old exams)
  - Can ignore register allocation
  - Can ignore automatic parallelization

Part I: Short Questions

- Fast questions – a couple of minutes each
- Basic facts/trends
- Most should be obvious, but some a little thought

Question 1 – Fall 2018

- What is the main difference between any-path and all-path dataflow analysis?

Question 3 – Fall 2018

- When a compiler scheduler wants to speculate an instruction, name one issue that it must consider to preserve correctness of the resulting code.

Question 4 – Fall 2019

- If any instruction in a basic block is scheduled later than its Lstart, then the schedule length for the basic block will be longer than the critical path length. Is the preceding statement True or False? Explain your answer

Question 2 – Fall 2020

- It is often possible to improve the performance of a loop limited by RecMII by adding resources to the processor. Is the preceding statement True or False? Justify your answer

Part II: Medium/Long Questions

- Longer questions
  - Problems that must be worked out: 5-10 mins each
  - Some question like lecture examples
  - Some question have a little twist
- Practicing problems ahead of time will make this smoother

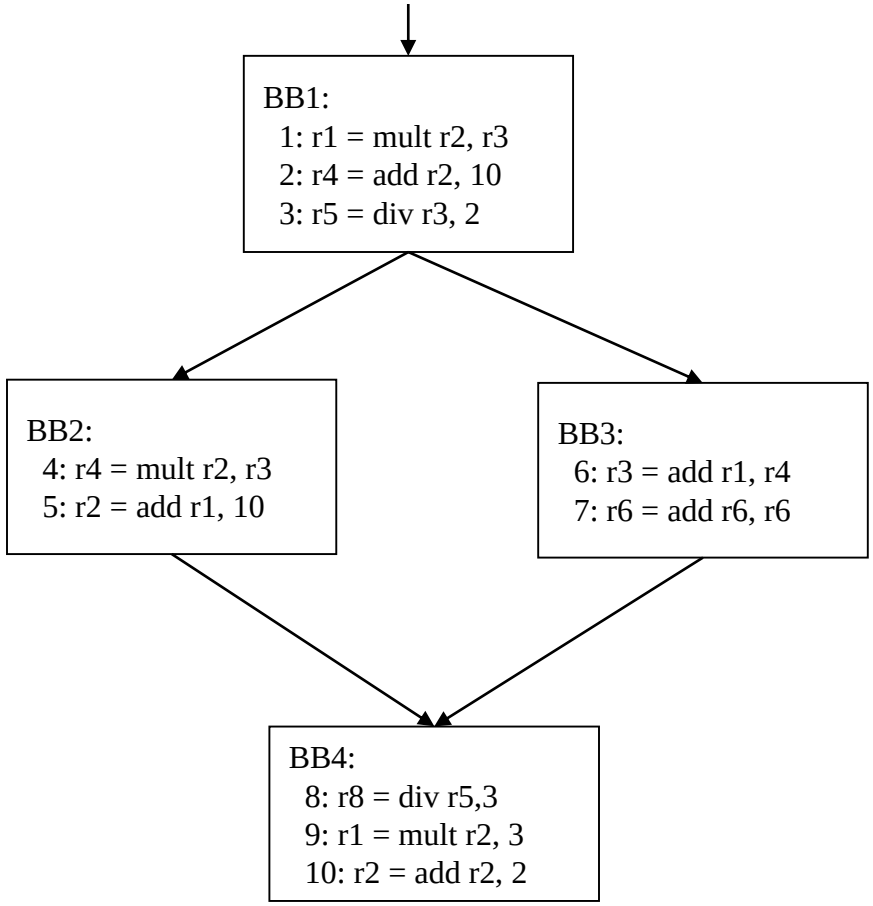
Question 6 – Fall 2019

Draw the control flow graph (CFG) and determine
the *minimum* number of predicates required to if-convert the following code.

```
do{
   if (a>0 && b>0){
      if (c>0)
         x+=1;
      else
         x+=2;
      z=x/3;
   }else{
      y+=1;
   }
}while(z<100)
```

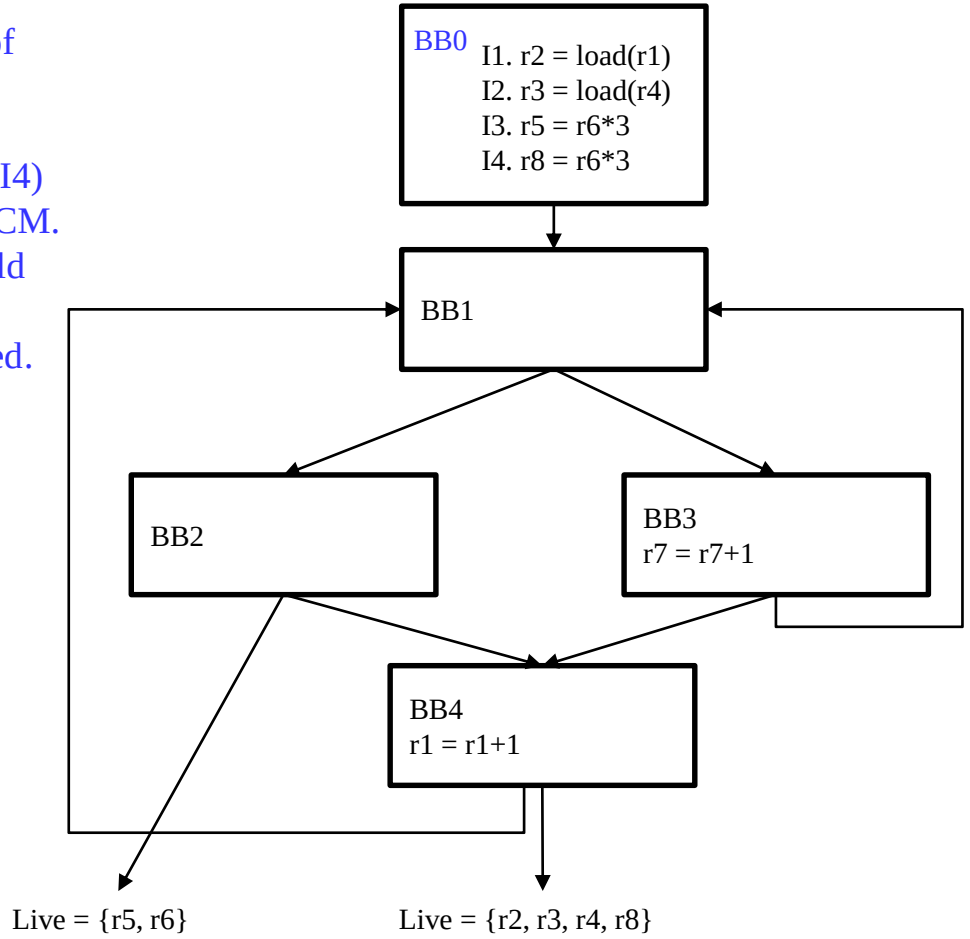
Question 7 – Fall 2020

Compute the Available Expression GEN/KILL/IN/OUT sets at BB4.
Assume r2, r3, r6 are defined before entering BB1



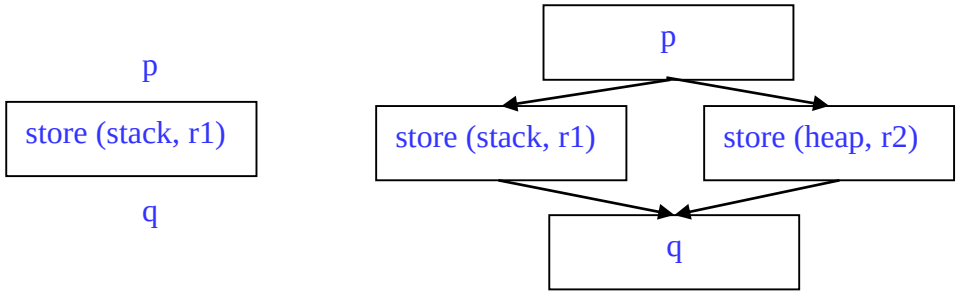
Question 8 – Fall 2019

You are trying to reverse engineer some optimized assembly code to determine the original locations of instructions before optimization. In the following loop consisting of 4 basic blocks (BB1-BB4), the preheader (BB0) contains 4 instructions (I1, I2, I3, I4) that were possibly removed from the loop using LICM. For each instruction, determine whether LICM could have been legally applied and if so, which basic block(s) the instruction could have originally resided.



Question 11 – Fall 2018

- You are building a new compiler where you will have no memory dependence analysis capabilities. The only intelligence that you have is that you can differentiate stack/heap accesses. To enable some optimizations of loads (i.e., CSE or LICM), you decide to build a dataflow analysis pass to summarize the presence of heap/stack store instructions to avoid scanning basic blocks repeatedly. Your dataflow is defined as follows: *A heap (stack) store is present if there exists at least one store to the heap (stack) starting from p and ending at q*. When no heap (stack) store is present, you will be able to freely optimize loads between p and q.



Question 11 (continued)

- (a) Is this a forward or backward dataflow analysis problem?
- (b) Is this an all-path or any path dataflow analysis problem?
- (c) Define GEN and KILL sets to compute store presence. *Hint: Consider the use of special variables: heap and stack*

Blank Slide

Blank Slide